WHAT MAKES FOR A HIT POP SONG? WHAT MAKES FOR A POP SONG?

NICHOLAS BORG AND GEORGE HOKKANEN

Abstract. The possibility of a hit song prediction algorithm is both academically interesting and industry motivated. While several companies currently attest to their ability to make such predictions, publicly available research suggests that current methods are unlikely to produce accurate predictions. Support Vector Machines were trained on song features and YouTube view counts to very limited success. We discuss the possibility that musical features alone cannot account for popularity. Current research into automated genre detection given features extracted from music has shown more promising. Using a combination of K-Means clustering and Support Vector Machines, as well as a Random Forest, we produced two automated classifiers that performs five times better than chance for 10 genres.

1. Introduction

2. Data

Our main source of data was the Million Song Dataset Subset, distributed by Labrosa. The subset provides pre-extracted features for 10000 songs, including: song and artist names, song duration, timbral data (MFCC-like features) and spectral data for the entire song, number of sections and average section length, and a number of other features.

In order to measure the popularity of a song, for each song we collected the number of view counts registered on the video returned as the first link in a YouTube search using the YouTube API, the query being the song's and artist's names. We checked by hand the accuracy of this scraping method and concluded that the two errors in thirty randomly drawn songs was not a problem given that the errors also coordinate well with very low view counts (i.e. something unpopular enough to not return a copy of the song on youtube ends up returning an unrelated video with a low view count).

3. Preliminary Analysis

We first ran basic correlation coefficients between different parts of the metadata and also with our extracted youtube view counts. The results were largely insignificant and included weak correlations such as one of .2 between the tempo and loudness metadata features. Correlations between the youtube view counts and the echonest metadata features loudness, tempo, hotttness, and danceability were completely negligible (less than .05 in magnitude). The fact that these are not at all correlated is interesting in its own right because it points to no single metadata feature being at all a good predictor of views on youtube

4. Popularity Prediction

4.1. **Linear Classification using Support Vector Machines.**

4.2. **String Kernel.** Given that the mfcc and spectral data is temporal, we wanted to use the ordering therein to describe the sound. Motivating a string kernel SVM approach, we create 'string' features for our songs as follows. For each i, we take the spectral bucket i (corresponding to a frequency-aggregate magnitude) for each spectra vector within a range of each song (usually about 45 seconds in the middle). This gives a list of values which correspond to the magnitudes over time of this slice of the sound spectrum for every song. Next, using a subset of this data (we used two hundred of the ten thousand songs) we compute a list of intervals (we used 26 of them, corresponding to the characters a through z) that uniformly distribute the data. Then using these intervals, we compute a string for each sequence of data obtained in the first step by replacing each value with a symbol or letter that represents the interval.

5. Popularity Prediction Results

6. Genre Classification

7. Genre Classification Results